A

Topic approval Seminar

on

Intelligent System to Predict Workload of Container-Based Microservices

Submitted in partial fulfilment of the requirement for the degree of PhD

In

Science and Technology

Computer Engineering

By

Name of Research Scholar: Mr. Prashant Bharat Sawant

Registration No: 2220220340

Under the guidance of

Dr. Hariram Chavan

Department of Computer Engineering

K J Somaiya Institute of Technology

Somaiya Ayurvihar Complex Eastern Express Highway Near Everard Nagar, Sion East, Mumbai, Maharashtra 400022

मुंबई विद्यापीठ
University of Mumbai

2022-23

CERTIFICATE OF APPROVAL

This is to certify that Name of the Student has delivered a seminar on **"Intelligent System to Predict Workload of Container-Based Microservices"** on date August 31,2023 and submitted a report in **K J Somaiya Institute of Technology** Sion Mumbai, in the partial fulfilment of the degree of PhD in the Science and Technology **Computer Engineering.**

Dr. Hariram Chavan
Guide

Dr. Mahesh Shirorle
Head Dep of Comp & IT
Subject Expert External

Dr. Sunita Patil
Vice-Principal
Subject Expert Internal

Dr. Sarita Ambadekar
Head of the Department

K J Somaiya Institute of Technology Sion Mumbai 400 022

23.09.23
Dr. Hariram Chavan
PhD Coordinator

Dr. Suresh Ukarande
Principal

Principal
K J Somaiya Institute of Technology
Sion (E), Mumbai - 400 022

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sawant

Mr Prashant Bharat Sawant

Registration No: 2220220340

Date: 25/09/2023

Place: Mumbai

ii

ACKNOWLEDGEMENT

A few sublime human experiences defy expressions of any kind, and a feeling of true gratitude is one of them. we, therefore, find words quite inadequate to express my indebtedness to my Guide and PhD coordinator **Dr. Hariram Chavan**, his virtuous guidance, encouragement and help throughout this work. Their deep insight into the problem and the ability to provide solutions has been of immense value in improving the quality of work at all stages. This experience of working with them shall ever remain a source of inspiration and encouragement for me.

I express my thanks to **Dr. Sunita Patil, Vice Principal KJSIT**, Sion for extending her support that she gave to truly help the progression of the project work.

My sincere thanks to **Dr. Suresh Ukarande Principal, KJSIT**, Sion for providing us the necessary assistance in the completion of the work.

I am extremely grateful to the celebrated authors whose precious works have been consulted and referred to in my work. I also wish to convey my appreciation to my friends who provided encouragement and timely support in the hour of need.

Special thanks to my Parents and family whose love and affectionate blessings have been a constant source of inspiration in making this a reality.

All the thanks are, however, only fraction of what is due to Almighty for granting me an opportunity and the divine grace to successfully accomplish this assignment

Bawant

Mr Prashant Bharat Sawant

Registration No: 2220220340

iii

Abstract

The adoption of Cloud Native Applications has significantly expanded as a result of the explosive growth of communication technologies, IoT, and cloud computing. This native application's flexibility during deployment in a production setting help in lowering server costs. AWS, GCP, and other cloud servers are where the majority of these Cloud Native Applications are created and deployed. Edge computing is gaining popularity for a number of reasons, including avoiding latency, Edge computing is gaining popularity for a number of reasons, including the necessity for autonomous disconnected operations, privacy, and security as well as the desire to avoid latency, data volume, and bandwidth constraints. These factors are driving organizations in all sectors to operate an increasing number of applications at edge locations. These programs have typically been run on virtual machines (VMs), but with the advent of Kubernetes and microservices architectures, more apps are being containerized, necessitating the use of a cloud native computing stack at the edge.

The demand for microservices has surged as a result of the growing popularity of containers over virtualization. Managing many containers in the production environment becomes difficult. Existing work on container management is based on Kubernetes Security, Network, Storage & resource placement of containers, deduplication of docker images. Still there is scope to improve prediction of load while using containers with microservices. Many research works have used hybrid model to predict load on containers. During monitoring load parameters like CPU & memory are used most of the time. Most of the research are underperformance when there is sudden spike in workload. In this research work, we are attempting to investigate additional factors called it as Volatility index along with CPU, memory and other parameters, which have a direct impact on the performance of containers while employing microservices. Additionally edge node doesn't have sufficient resources to process client request. In Kubernetes all the information of node are store in etcd, which act as database for entire Kubernetes framework. This etcd is responsible for maintaining & delivery node information to kube controller. In large cluster it create bottleneck in etcd. Which directly create impact while delivering services This research work is trying to solve this bottleneck problem so that services can be delivered effectively.

Table Contents

Chapter 1: Introduction 3
1.1 Service Model 3
1.2. Deployment Models 3
1.3. Virtualization vs Containerization 4
1.4 Microservices 5
Chapter 2: Literature Review 7
2.1 Background Work 7
2.2 Kubernetes 8
2.3 Container 9
2.4 multi-Cloud 9
2.5. Research Gap 9
Chapter 3: Objective 10
Chapter 4: Methodology 11
Chapter 5: Conclusion 13
References 14

List of Figures

Content	Page No
Fig 1.1 Virtualization vs Containerization	4
Fig 1.2 Architecture of Docker.	5
Fig 1:3 Microservice vs Monolith Architecture.	5
Fig 2.1: Summary of the filtering process to obtain the set of 105 peer-reviewed	8
Fig 4.1: Load Prediction using Hybrid Model & Volatility Index	11
Fig 4.2: Alert Notification Using Volatility Index	12

Chapter 1: Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing help many businesses, government and organization to improve the business, because nature of service offered by cloud computing model. Since last two-three decade there is tremendous growth in use of Cloud Computing services.

Following are the services offer by Cloud Model.

1.1 Service Model

SaaS: Software as a Service, Customers are provided with applications that are accessible anytime from anywhere. Resource management of Application is taken care by provider. Google Drive, Gmail, Facebook or web-based application is example of software as a service.

PaaS: Platform as a Service, Customers are provided with a platform for developing applications hosted in cloud. User don't need to focus on configuration of Application. Platform will be provided from Provider.

IaaS: Infrastructure as a Service, Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure. In IaaS application which is running on infrastructure is completely managed by user only not provider. Provider is only maintaining hardware.

1.2. Deployment Models

Private Cloud

Infrastructure access within the premises. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers. All the important data in maintained in own datacentre, which is very important whenever we are considering security. In private cloud infrastructure is purchased by institute/organization and to maintain this infrastructure they need to hire high skill person.

Public Cloud

In public cloud resource are shared by publicly. AWS, Azure these are the example of public cloud. The cloud infrastructure is provisioned for exclusive use by a single organization

comprising multiple consumers to public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.

1.3. Virtualization vs Containerization

Virtualization-based technologies have become ubiquitous in computing. While they provide an easy-to implement platform for scalable, high-availability services, they also introduce new security issues. Traditionally, discussions on security vulnerabilities in server platforms have been focused on stand-alone environments. For cloud and virtualized platforms, the discussion focuses on the shared usage of resources and the lack of control over the infrastructure Since last two-decade cloud computing is dominating market.

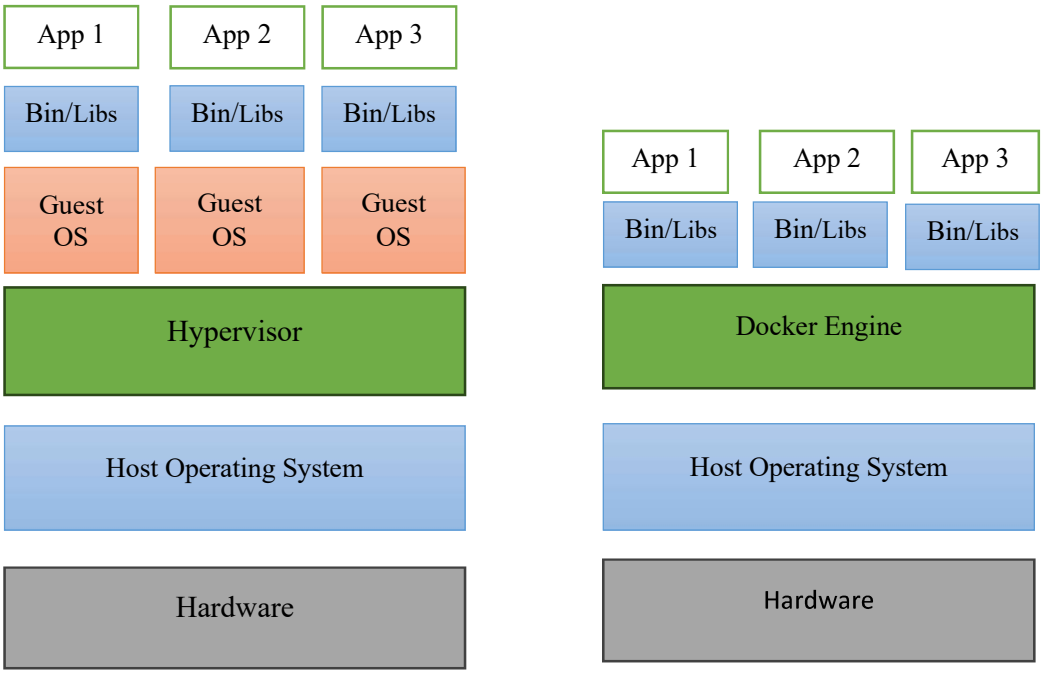

Fig 1.1: Virtualization vs Containerization

Docker [13] is open-source platform that provide facility to developer to automate the deployment of application, scaling, and management of applications inside lightweight, portable containers. These containers are isolated environments that package an application and its dependencies, ensuring consistency and reproducibility across different environments. Docker Engine: The essential element in charge of managing containers. To communicate with Docker, it has a server, REST API, and command-line interface (CLI).

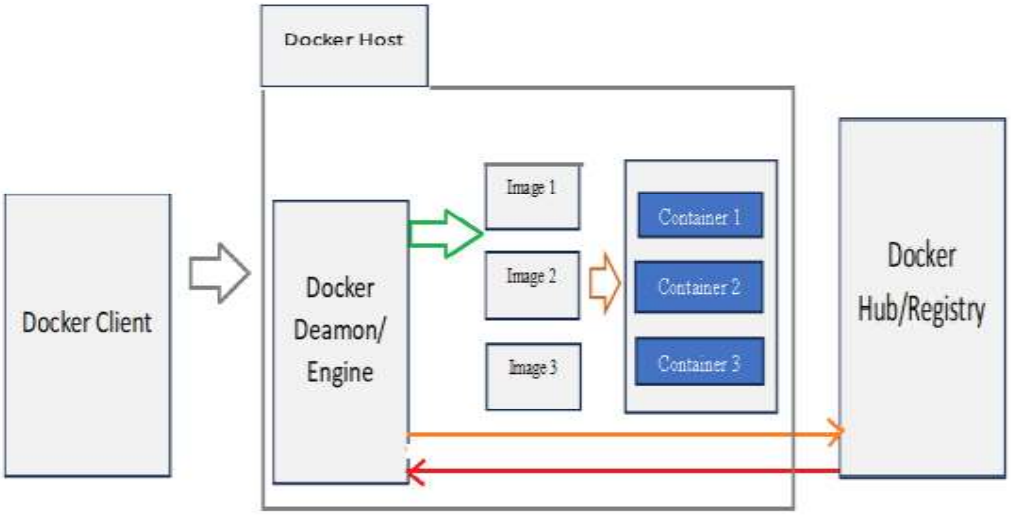

Fig 1.2: Architecture of Docker [13]

In fig 1.2 Images which are the one of the components of docker which are read-only templates that contain an application and its dependencies. Another component is Containers which are instances of Docker images, running in an isolated environment. They are lightweight and share the host OS's kernel, making them fast and efficient. Docker client communicate with docker engin through commands. Container cane be created using Docker file which are a text file containing a set of instructions to build a Docker image. It specifies the base image, dependencies, environment variables, and other configurations. Docker Hub which is a public repository of Docker images, where you can find and share pre-built images for various applications and services.

1.4 Microservices

Microservices is an architectural style and approach to building software applications as a collection of loosely coupled, small, and independent services. Each service in a microservices architecture focuses on a specific business capability and runs as a separate process, communicating with other services through well-defined APIs. These services work together to form a larger application, each responsible for a specific function or feature.

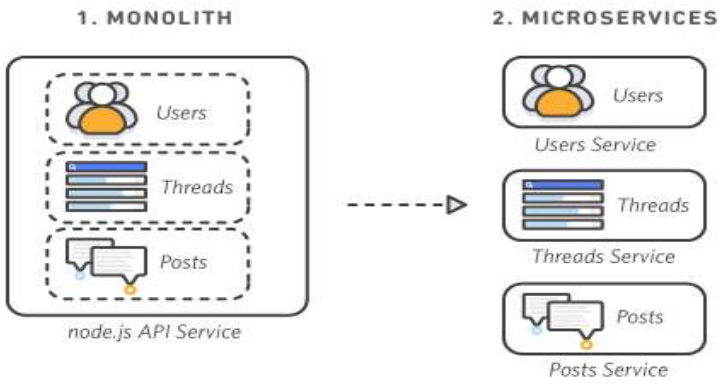

Fig 1.3: Microservice vs Monolith Architecture [21]

1.5 Edge Computing

Much of today's computing already happens at the edge in places like hospitals, factories and retail locations, processing the most sensitive data and powering critical systems that must function reliably and safely. These places require solutions with low latency that do not need a network connection. What makes edge so exciting is the potential it has for transforming business across every industry and function, from customer engagement and marketing to production and back-office operations. In all cases, edge helps make business functions proactive and adaptive—often in real-time—leading to new, optimized experiences for people.

Edge allows businesses to bring the digital world into the physical. Bringing online data and algorithms into brick-and-mortar stores to improve retail experiences. Creating systems that workers can train and situations where workers can learn from machines. Designing smart environments that look out for our safety and comfort. What these examples all have in common is edge computing, which is enabling companies to run applications with the most critical reliability, real-time and data requirements directly on-site. Ultimately, this allows companies to innovate faster, stand up new products and services more quickly and opens up possibilities for the creation of new revenue streams.

Chapter 2: Literature Review

Virtualization technology is heart of Cloud computing. When opposed to virtual machines, containerization is a technology used in the cloud that is growing in popularity because of benefits such being lightweight, scalable, and available. With the assistance of businesses like Google, Amazon, the Docker container is expanding quickly and is frequently utilized in large-scale production cloud environments. This section we will discuss about background work on Resource management of Container based application.

2.1 Background Work

Liangkang zhand et al [1] and et all mentioned issues with container workload. They considers the hybrid model for prediction of container workload. They have used LSTM & Tripple Exponential smooth model for work load prediction. Most of the work is done on efficient resource allocation in cloud environment.

Yulai Xie et al [2] worked on Predicting the resource usage of container workload using ARIMA with triple exponential smoothing to accurately predict both the linear and non-linear relationships in the time series of the resource workload of Docker container.

Containers are created from images which are shared between users via a registry. The amount of data registries store is massive. Nannan zhao [3] discussed image characterization like properties of layers and images in terms of size, compression ratio, popularity. He studies deduplication of image using layer sharing, file level deduplication. Nannan zhao[3] investigated the I/O performance of image retrievals and storage drivers and drew several design implications. parallel compression algorithms such as Pig zand Lz4can significantly reduce compression/decompression time. He mentioned there is need to do analysis of HPC based container workload.

Kejiang et al[4] designed Hadoop cluster on container. Spark application on four containers. Tried to understand the performance running application on these containers [4]. Kejiang et al[4] have used Modelling techniques over CPU, memory and I/O parameters to characterize the performance of applications running in containers. Author Identified parameters like CPU, disk I/O, Memory which affect the performance of application. Using these parameters, he tested Hadoop based Spark application on container using SVM, LR [4]. He concludes that SVM & ANN gives a better result that LR.

According to Mubin Ul Haque et al[5] their empirical study over 113,922 posts from the SoF community to understand interests and challenges faced by software developers with respect to Docker technology. Author Conclude with most of the user are facing Networking Error and Memory Management are more challenging to the community.

There are many open-source monitoring tools [18,19,20] are available in market that can be used to monitoring log of containers. These tools help for system administrator to take effective action against incident.

2.2 Kubernetes

Kubernetes is an open-source container orchestration system that is used to deploy, scale and manage application containers automatically. Developed by Google, Kubernetes is a platform that is only accessible via a domain-specific Representational State Transfer Application Programming Interface and supports a wider range of clients by applying higher-level versioning, validation, semantics, and policy. The system works with many containerising tools including Docker. It helps manage containerised applications in different deployment environments such as physical virtual or cloud environments. It offers many features such as high availability, scalability and disaster recovery

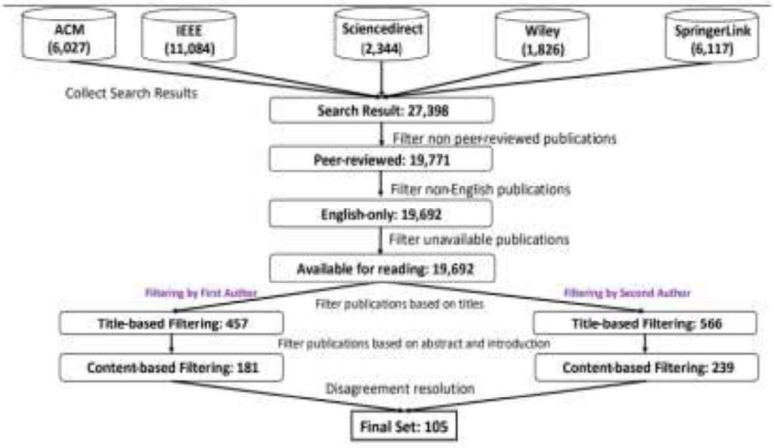

Fig 2.1: Summary of the filtering process to obtain the set of 105 peer-reviewed

In fig 2.1 researcher studied almost 105 peer reviewed, articles, official documents of Kubernetes. Most important thing is that in his study researcher cover benefits & Challenges of of Kubernetes. Shazibul et al [11] have mentioned networking, storage, resource allocation issues in their study.

2.3 Container

Based on the literature review we can say that Deployment of any application on container or on serverless technology is challenging, because we need to do closely watch with resource placement. Microservice based application are more scalable and maintainable due to this most of the industry are switched to cloud native development.

2.4 multi-Cloud

Shazibul [11] explain framework for policy-based application deployment management in the multi-cloud environment. Deployment of microservices in multi cloud is challenging. By looking at cost many vendor are preferring multi cloud for their business.

2.5. Research Gap

Based on the literature survey we found that docker is widely used in production environment. It's important to maintain application in healthy environment. We found that most of the researcher have used machine learning & Deep Learning model to resolved issues related to resources allocation, container placement, schedular, networking. Still there is scope to work on prediction of container based microservices. Many research have worked on orchestration tools like Kubernetes.

1. Investigate hybrid model [1] with other parameters.
2. It is necessary to find out the reason behind the mechanism by which the demand for services increases and decreases immediately
3. Lack of resources in edge node, so it's difficult to process request at node.
4. In Kubernetes data is store in etcd. So in large cluster it create bottleneck, which directly affect at the time of delivery of services.

Chapter 3: Objective

As we have discussed in previous chapter [2] use of cloud-based application extensively increased a lot. Many organizations are moving to the multi-cloud-based model. It is important to explore container based microservices approach in multi-cloud environment also. By looking at the current challenges in container-based microservices we will try to resolve these issues by consider following objective.

1. To Design Efficient Microservices Deployment Strategy.
2. To Measure Volatility Index for Work Load of Container
3. To Design Hybrid Model to Predict work Load of Container Based Microservices
4. To Mitigate Bottleneck of Kubernetes Database.

Chapter 4: Methodology

In software Industry most of the application are hosted on Cloud Server. Cloud Provider are taking care of those servers. After deploying application on container, it is important to monitor its resources utilization. There are n number of containers running on single host. To run any application on container smoothly, it requires resources for execution. To manage this application which are running on multiple Containers need to be orchestrated using swarm or Kubernetes. If any Container is not getting any resource, then this Container get died. In this situation orchestration tool help us to scale up and scale down Container, so that desired state of cluster can be maintained. But, it's important to understand and analyse behaviour of these Containers. These Behaviour can be analysed by monitoring, vcpu, i/o rate, memory, Network

Multiple containers are sharing host machine resources. So, there is chance of bottleneck, while allocating resources. Even though multiple containers are running, only few Containers are having continuous load. Rest of Container are free. So it is important to learn container work load pattern. Based on the work load, we can take proactive action before those Containers died.

In our research work, we are trying to resolve this problem by using machine leaning model. This machine learning model will predict load on container. Accordingly, we can take action if more resources are required for any Container.

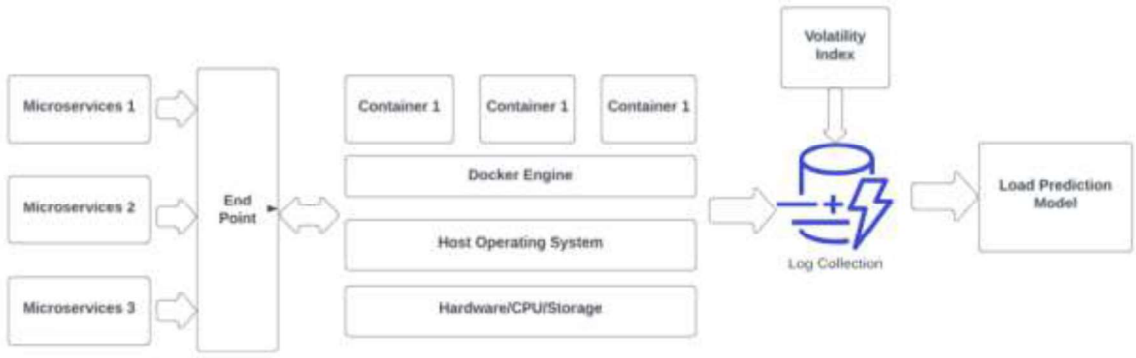

Fig 4.1: Load Prediction using Hybrid Model & Volatility Index

We are using Hybrid model which will consider parameter like cpu, memory, storage, network packets etc. Along with that volatility. Which help to give alert to system.

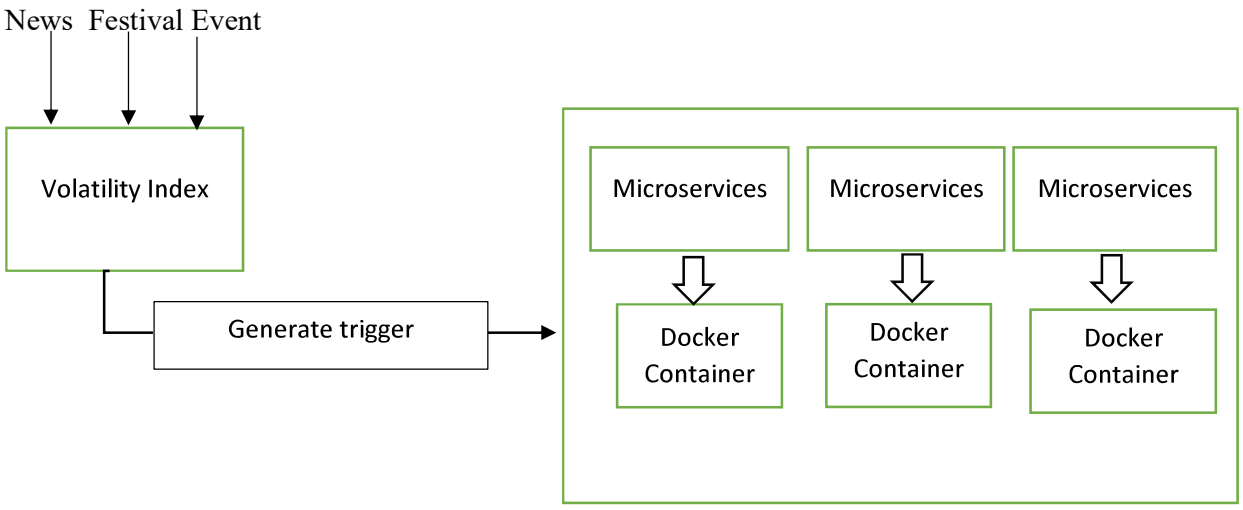

Fig 4.2: Alert Notification Using Volatility Index(VI)

In fig 4.2 volatility index(VI) is consider as a parameters which will consider News, Festival, Event, attack etc Based on this parameter volatility will changes. Simultaneously this VI will send notification to the system so that system can maintain resources accordingly.

Chapter 5: Conclusion

In this study, we identified possible issues of container based microservices. Due to lightweight & portability feature of containers they are widely used in production environment for deployment. So it is important to monitor work load of this microservices. Existing work on containerization technology is focused on security, networking & container load balancer including Kubernetes. Yulai, Zhang , Nannan [1] [2][3] worked on workload prediction of containers. They have used only two parameters which includes CPU & memory. But their proposed solution ARIMA & LSTM not able to handle sudden spike arises on containers. Due to this lag in containers performance arises. In this study we trying to overcome these issues by identifying root cause of this sudden spike, which can be measure in terms of volatility index. This volatility index help to containers to prepare for workload. To managed this container orchestration tools like Kubernetes are used, which are maintain cluster information in databases know as etcd. Whenever there are multiple request are coming from large cluster, these request are forwarded to etcd, which centralized database. In this case there is bottleneck issues are created. This research work is trying to

Edge computing application are growing day by day because of enhancement in IoT, communication technology like 5G. But due to less resource are available at edge node it difficult to process data at their end. This study will try to resolve this problem by using container based microservices. This study will help to decide optimal strategy to deploy microservices using AI/ML

References

[1] Liangkang Zhang, Yulai Xie,“A Novel Hybrid Model for Docker Container Workload Prediction”, IEEE Transactions On Network And Service Management, Vol. , February 2023

[2] Yulai Xie, Dan Feng “Real-Time Prediction of Docker Container Resource Load Based on a Hybrid Model of ARIMA and Triple Exponential Smoothing”, IEEE Transactions On Cloud Computing, Vol. 10, No. 2, April-June 2022

[3] Nannan Zhao “Large-Scale Analysis of Docker Images and Performance Implications for Container Storage Systems”, Ieee Transactions On Parallel And Distributed Systems, Vol. 32, No. 4, April 2021

[4] Kejiang Ye, Yanmin Kou, “Modeling Application Performance in Docker Containers using Machine Learning Technique”, Modeling Application Performance in Docker Containers using Machine Learning Technique

[5] Mubin Ul Haque, Leonardo Horn Iwaya, “Challenges in Docker Development: A Large-scale Study Using Stack Overflow”, Conference’17, July 2017,ACM

[6] Joy Rahman ,“Predicting the End-to-End Tail Latency of Containerized Microservices in the Cloud” 2019 IEEE International Conference on Cloud Engineering (IC2E)

[7] Amit Podar, Naryan DG, “Performance Evaluation of Docker Container and Virtual Machine”

[8] Deployment Management and Topology Discovery of Microservice Applications in the Multicloud Environment”, Journal of Grid Computing, Published: 27 January 2021

[9] Sourav Kanti Addya ,“CoMCLOUD: Virtual Machine Coalition for Multi-Tier Applications Over Multi-Cloud, IEEE Transactions On Cloud Computing, Vol. 11, No. 1, January-March 2023 Environments”

[10] Lei Che “Docker Container Log Collection and Analysis System Based on ELK”, 2020 International Conference on Computer Information and Big Data Applications (CIBDA)

[11] Shazibul Islam Shamim,Benefits, Challenges, and Research Topics: A Multi-vocal Literature Review of Kubernetes, arXivLabs,Cornell University.

[12] J. N. Acharya and A. C. Suthar, “Docker container orchestration management: A review,” in Proceedings of the International Conference on Intelligent Vision and Computing (ICIVC 2021), H [13] Sharma, V. K. Vyas, R. K. Pandey, and M. Prasad, Eds. Cham: Springer International Publishing, 2022, pp. 140–153

[14] Nima Mahmoudi, “Performance Modeling of Metric-Based Serverless Computing Platforms”, IEEE Transactions On Cloud Computing, Vol. 11, No. 2, April-june 2023

[15] Ye and Y. Ji, “Performance tuning and modeling for big data applications in docker containers,” in 2017 International Conference on Networking, Architecture, and Storage (NAS). Los Alamitos, CA, USA: IEEE Computer Society, aug 2017, pp. 1–6.

[16] J. N. Acharya and A. C. Suthar, “Docker container orchestration management: A review,” in Proceedings of the International Conference on Intelligent Vision and Computing (ICIVC 2021), H. Sharma, V. K. Vyas, R. K. Pandey, and M. Prasad, Eds. Cham: Springer International Publishing, 2022, pp. 140–153

[17] Al-Doghman, Firas, Nour Moustafa, Ibrahim Khalil, Zahir Tari, and Albert Zomaya. "AI-enabled secure microservices in edge computing: Opportunities and challenges." IEEE Transactions on Services Computing (2022).

[18] Ismail, Bukhary Ikhwan, Ehsan Mostajeran Goortani, Mohd Bazli Ab Karim, Wong Ming Tat, Sharipah Setapa, Jing Yuan Luke, and Ong Hong Hoe. "Evaluation of docker as edge computing platform." In 2015 IEEE conference on open systems (ICOS), pp. 130-135. IEEE, 2015.

[19] Zhang, Jiqiang, and Pengcheng Liu. "Application of edge computing in intelligent security system." In Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics, pp. 216-220. 2022.

[20] https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf

[18] https://prometheus.io/

[19] https://grafana.com/

[20] https://docs.docker.com/get-started/overview/

[21] https://microservices.io/patterns/microservices.html